PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : H04J 3/12, 3/24	A1	(11) International Publication Number: WO 92/21187
		(43) International Publication Date: 26 November 1992 (26.11.92)

(21) International Application Number: PCT/US91/03448

(22) International Filing Date: 23 May 1991 (23.05.91)

(71) Applicant *(for all designated States except US)*: BELL ATLANTIC NETWORK SERVICES, INC. [US/US]; 1310 North Court House Road, Arlington, VA 22201 (US).

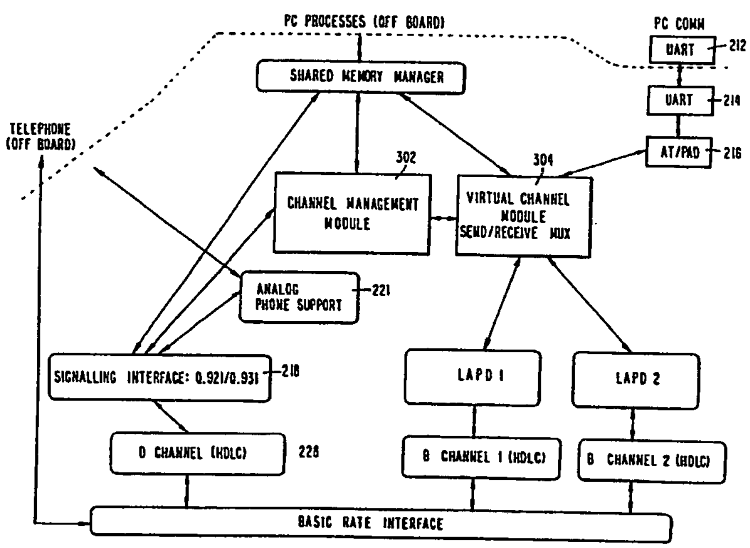
(72) Inventor; and
(75) Inventor/Applicant *(for US only)* : KO, Cheng, Hsu [US/US]; 5 Manor Parkway, Salem, NH 03079 (US).

(74) Agents: BECKER, Stephen, A. et al.; Lowe, Price, LeBlanc, Becker & Shur, 99 Canal Center Plaza, Suite 300, Alexandria, VA 22314 (US).

(81) Designated States: AT, AT (European patent), AU, BB, BE (European patent), BF (OAPI patent), BG, BJ (OAPI patent), BR, CA, CF (OAPI patent), CG (OAPI patent), CH, CH (European patent), CI (OAPI patent), CM (OAPI patent), DE, DE (European patent), DK, DK (European patent), ES, ES (European patent), FI, FR (European patent), GA (OAPI patent), GB, GB (European patent), GR (European patent), HU, IT (European patent), JP, KP, KR, LK, LU, LU (European patent), MC, MG, ML (OAPI patent), MR (OAPI patent), MW, NL, NL (European patent), NO, PL, RO, SD, SE, SE (European patent), SN (OAPI patent), SU, TD (OAPI patent), TG (OAPI patent), US.

Published
With international search report.

(54) Title: DYNAMIC CHANNEL ALLOCATION METHOD AND SYSTEM FOR INTEGRATED SERVICES DIGITAL NETWORK



(57) Abstract

Dynamic channel allocation in an ISDN line is implemented above a Level II link layer protocol (LAP-D). The preferred embodiment comprises programming operating on a computing platform system including a communication coprocessor system. The programming includes a channel management module (CMM) (302) and a virtual channel module (VCM) (304) to implement dynamic channel allocation. The modules cooperate to control bandwidth between communication partners by selective allocation and deallocation of virtual B channels between them in response to preassigned priorities and real time events. Channel deallocation also takes place in response to high error rates in message transmission. The coprocessor system preferably is circuit board mounted and installed in an available slot of a personal computer.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	FI	Finland	ML	Mali
AU	Australia	FR	France	MN	Mongolia
BB	Barbados	GA	Gabon	MR	Mauritania
BE	Belgium	GB	United Kingdom	MW	Malawi
BF	Burkina Faso	GN	Guinea	NL	Netherlands
BG	Bulgaria	GR	Greece	NO	Norway
BJ	Benin	HU	Hungary	PL	Poland
BR	Brazil	IE	Ireland	RO	Romania
CA	Canada	IT	Italy	RU	Russian Federation
CF	Central African Republic	JP	Japan	SD	Sudan
CG	Congo	KP	Democratic People's Republic of Korea	SE	Sweden
CH	Switzerland			SN	Senegal
CI	Côte d'Ivoire	KR	Republic of Korea	SU	Soviet Union
CM	Cameroon	LI	Liechtenstein	TD	Chad
CS	Czechoslovakia	LK	Sri Lanka	TG	Togo
DE	Germany	LU	Luxembourg	US	United States of America
DK	Denmark	MC	Monaco		
ES	Spain	MG	Madagascar		

DYNAMIC CHANNEL ALLOCATION METHOD AND SYSTEM FOR INTEGRATED SERVICES DIGITAL NETWORK

Technical Field

This invention relates generally to Integrated Services Digital Network (ISDN), and more particularly to controlling dynamic channel allocation in ISDN and to creating a virtual channel out of underlying network channels.

Background Art

A digital telecommunication network can be characterized by two general features, the existence of well defined digital interfaces to Customer Premise Equipment (CPE) which are the end nodes of the network, and an interconnected switching backbone network which is capable, upon demand, of creating point-to-point connections between any two end nodes. An example of such a network architecture can be found in the ISDN specifications which have been developed by the International Telegraph and Telephone Consultative Committee (CCITT). Although aspects of the invention may be relevant to other types of digital telecommunication networks, in the paragraphs below, the general features of a digital telecommunication network upon which this invention depends will be described using examples drawn from ISDN.

A complete description of the architecture of ISDN is beyond the scope of this specification. For details, and for an extensive bibliography of references of ISDN, see Stallings, ISDN, An Introduction, MacMillan Publishing Company, 1989. In

general, all ISDN interfaces to customer premises equipment (CPE) can be characterized by the use of time division multiplexing at the physical layer to partition the bandwidth of a single physical link into multiple independent channels, such as the two B channels and the D channel found in the Basic Rate Interface (BRI). The B, or "bearer", channel is used to transmit user information. The D channel carries signalling or control data which provides control over the use of the two B channels in the BRI as well as user data which often is in the form of packets. These channels can be considered as independent ISDN resources available to the end user. ISDN also defines a specific data link layer and network layer protocol to be used over the D channel between the CPE and the Local Exchange (LE).

The B channel in an ISDN network is a full duplex, 64000 bit per second, synchronous user channel. There are many modes of operation for the B channel. The preferred embodiment of this invention is described in the environment of the BRI and is based in part on the capability of the B channel to carry digital data or PCM-encoded digital voice in a transparent manner across the circuit switched backbone network. The same principles could be extended to the Primary Rate Interface (PRI). Before a circuit switched B channel can be used by end users, the channel must be connected through the ISDN. Requests for channel connection are made by the CPE to the LE using a protocol over the D channel. The LE routes the request through the ISDN using other protocols, until the destination LE is reached. The destination LE then issues a call connection request to the destination CPE which can accept the call. Either

party to a call can disconnect the call. This action also involves relaying a series of messages over the respective D channels and other network activity. The ISDN network itself can also disconnect the call due to error conditions inside the network. Unlike analog telecommunication networks, the time to connect or disconnect a channel is anticipated to be very short, potentially on the order of a second.

An important characteristic of B channels in an ISDN is that the network, is permitted to create independent paths inside the network for each circuit switched B channel. Thus if a CPE connects two B channels to another CPE, the CPE can not rely on the network to synchronize the transmissions across the two channels. A second important characteristic of B channels in an ISDN is that they can be used for multiple purposes, such as voice service or data service, and a BRI can support multiple devices such as a computer and a telephone.

When one considers using the data services of ISDN to connect computers, the bandwidth of a single B channel is a serious limitation. This is particularly true when one considers that many computer network applications are geared toward local area networks in which applications commonly achieve point-to-point communication speeds on the order of several hundred thousand bits per second or greater. Another disadvantage of the use of circuit switched B channels is the bursty nature of data traffic and the fact that the tariff structure for circuit switched channels is independent of the data traffic over the channel. This can lead to inefficient use of a channel resulting in excessive channel usage cost.

An additional limitation is the desire to share the ISDN interface with other devices, such as a telephone, without having to statically allocate a channel to each user of the interface. In the event of contention for a channel between different users, high priority channel users should be able to preempt channels from low priority channel users. The present invention addresses these limitations and provides a method to overcome them.

Dynamic channel (bandwidth) allocation per se is not new to ISDN. U.S. Pat. No. 4,763,321 and U.S. Pat. No. 4,805,167 describe techniques to accomplish channel synchronization at the physical level (bit level synchronization) by manipulating the ISDN interface frame structures (TDM frames) to create a high data rate virtual channel. However, it is disadvantageous to carry out operations at the physical level (Level 1 of OSI) which require manipulation of hardware. The present invention does not operate at the physical level and makes no attempt to manipulate the ISDN physical interface.

Disclosure of Invention

Accordingly, a broad object of the invention is to improve bandwidth utilization of ISDN.

Another object is to dynamically allocate B channels between ISDN communication partners in response to preassigned priorities and real time events.

Another object of the invention is to enable computer networking on ISDN while retaining support of traditional telephone services.

An additional object is to carry out automatic ISDN channel allocation in a manner that is transparent

to a user.

Steill amother object of the invention is to carry out improved protocol for implementing dynamic bandwidth allocation in ISDN.

A further object of the invention is to carry out the aforementioned objects without manipulation of data at the physical layer.

To achieve the above and other objects, the invention provides a computer program implementation that enables a CPE device to dynamically manage the ISDN resources available to it in such a manner as to overcome the above mentioned limitations of ISDN. The invention enables ISDN to accommodate a computer network while retaining the ability to also support traditional telephony services.

One aspect of the invention carries out dynamic channel allocation in an ISDN line by monitoring events supplied to the line which indicate requests to use or release a B channel, evaluating channel usage priorities associated with users generating events monitored, and, based upon channel usage priorities, automatically setting up or tearing down B channels associated with the line. Processing takes place above LAP-D protocol.

In accordance with a preferred embodiment of the invention, upon a hardware platform are provided first and second program modules, a channel management module (CMM) and a virtual channel module (VCM), to implement dynamic channel allocation. The program modules cooperate to control bandwidth between communication partners by selective allocation and deallocation of virtual B channels between them in response to preassigned priorities and real time events.

The first program module, CMM, monitors requests

to connect and disconnect B channels by potential end users and the second module (VCM). The CMM is capable of signalling the VCM to force deallocation of a B channel from a virtual channel based on a request for channel connection by a higher priority end user. The CMM signals the VCM to indicate the existence of available B channels when end users disconnect the channel.

The second module, VCM, provides a block oriented transport service to various computer processes which can operate over one or more B channels. This service operates above the data link layer and manages the multiplexing of messages across one or more data link connections which operate on a B channel so as to ensure properly ordered delivery of message blocks. Channels are added or deleted on a dynamic basis, without interrupting the flow of data. The service supports the allocation and deallocation of channels based on signals from the CMM. Channel deallocation also takes place in response to high error rates in message transmission.

The program modules implemented in accordance with this invention provide control logic for a CPE device to allocate and deallocate ISDN channels to a plurality of end users, i.e., the local analog telephone equipment, the various processes running on the local computer system, and similar entities located remotely on the ISDN network. Implementation is based both on user specified priorities and the real time requirements of each user. The modules also support the virtualization of B channel resources so as to overcome the bandwidth limitations of a single B channel.

Still other objects and advantages of the present invention will become readily apparent to those skilled

in this art from the following detailed description, wherein only the preferred embodiment of the invention are shown and described, simply by way of illustration of the best mode contemplated of carrying out the invention. As will be realized, the invention is capable of other and different embodiments, and its several details are capable of modifications in various obvious respects, all without departing from the invention. Accordingly, the drawing and description are to be regarded as illustrative in nature, and not as restrictive.

Brief Description of the Drawings

FIGURE 1 is a circuit diagram of the hardware platform upon which programming in accordance with the invention is implemented.

FIGURE 2 is a system software module diagram of a communication coprocessor system known in the prior art and forming part of the hardware platform shown in FIGURE 1 for implementation of the present invention.

FIGURE 3 is a software module diagram corresponding to FIGURE 1 including additional software modules comprising an aspect of the invention.

FIGURE 4 is a message interface diagram for the additional modules shown in Figure 3.

FIGURES 5(a) and 5(b) are diagrams of B channel state tables and priority levels maintained by the CMM.

FIGURE 6 is a flow chart of programming carrying out an aspect of the invention.

FIGURE 7 is a representative view of a circuit board carrying the hardware shown schematically in FIGURE 1.

FIGURE 8 is a flow chart of programming for removing B-channels from the virtual channel in

response to an excessive error rate in accordance with another aspect of the invention.

FIGURE 9 is a flow chart of a routine for sending to a remote resource.

FIGURE 10 is a flow chart of a routine for receiving from a remote resource.

Detailed Description of the Invention

1. Overview

The preferred embodiment of the invention provides a computer-to-ISDN interface residing at customer premesis equipment (CPE) and operating above LAP-D protocol to perform dynamic channel allocation and deallocation in response to processes occurring on the ISDN and in accordance with user priorities. In the preferred embodiment, the CPE comprises a computer communication coprocessor board which contains an RJ-45 BRI interface (either at the ISDN T reference point or at the U reference point), an RJ-11 interface, a shared memory interface to an AT bus, and a UART interface to an AT bus. The interface is implemented by programming executed by a processor that, in the preferred embodiment, is mounted on a commercially available Telios B101 ISDN communications coprocessor board. The programming alternatively can be executed by other systems having the characteristics described below.

2. System Requirements

The system includes a real time operating system kernel capable of supporting multiple processes which execute as threads in a common address space. A message passing and signalling system permits various processes on the ISDN to communicate

with each other. The kernel supports voluntary blocking of a process until the occurrence of an event in the system and process rescheduling based on the occurrence of the event have taken place. (As used herein, an "event" is a hardware interrupt or the arrival of a message sent by another process in the system.) This type of system support is common to many real time operating system kernels such as PSOS, VRTX, or RTU.

Furthermore it is assumed that the system has an initial startup process which supports the loading of additional processes and their initialization, as well as initialization of various hardware devices in the system. The preferred embodiment of the invention is based on the PSOS kernel together with a capability for initialization and downloading other processes.

The system furthermore includes hardware interrupt routines which support plural hardware interfaces in the system and which use real time kernel message passing to cause a process to be executed to further service the event. The following hardware interrupt routines preferably are present in the system:

1. At least one interrupt routine to support HDLC level 2- operations on the two B channels and the D channel, such as flag detection and generation, zero bit insertion and deletion, CRC checking and generation, and address masking. Such a routine may operate with a DMA controller.

2. An interrupt routine to support a shared memory interface to a computer system bus, such as an ISA bus (AT bus). This routine, which may be in the form of a co-routine on the computer processor, must be able to route messages to all the processes in

the system.

3. An interrupt routine to support a UART interface to the ISA bus or other computer system bus.

4. The system furthermore should include a software module which implements the user side of the ISDN D channel signalling protocol. This includes the data link layer, that is, LAP-D, as specified in ISDN specification Q.921, and the network layer as specified in Q.931.

5. The system must be able to support an analog telephone, manage the conversion of analog to digital signals according to the ISDN standard, and interface to the program interrupt routines to support connection/disconnection of a voice call.

6. The system additionally must implement a data link layer protocol which can support a separate data link connection over each of the B channels. It is assumed that the partner for this data link connection is a similar process running on a similar ISDN communication coprocessor system. The data link process will have an implementation which supports full duplex operation by separate transmit and receive queues and interrupt driven thread dispatching based on the event of a message arriving to the system or a message being queued for transmission. In other words, the transmit and receive threads are non-blocking from a dispatch point of view.

7. The system of the preferred embodiment further should include a software module which implements a translator to map the AT command set into ISDN commands to connect and disconnect B channels. This module will also implement a PAD function to interface a byte oriented UART to a block oriented HDLC controller.

3. Platform Hardware

The hardware platform in the preferred embodiment is implemented by the commercially available Telios B101 ISDN Communication Coprocessor System board. The process identified as 4 in the preceding system requirements is carried out by Signalling Interface Process (SIP) software accompanying the Teleos B101. The capability identified as 5 is implemented by the VFI processes in Telios B101, and the function identified as 7 is implemented by its AT/PAD process. Further details of the Telios B101 communication coprocessor are given in the B101PC Hardware Reference Manual (REV A), 1990, available through Telios Communications Incorporated.

The present invention is implemented at customer premesis equipment, preferably by incorporation on at least one printed circuit board 700 as shown in FIGURE 7, adapted to be mounted in an available slot of a personal computer PC. A conventional analog telephone T is connected to the board 700 at and RJ-11 plug 702. The hardware platform upon which the invention is implemented, and as depicted schematically in FIGURE 1, includes an internal buss 100 supporting a microprocessor 102, preferably a 68000 type, such as a 68HC00CP-12). A direct memory access (DMA) controller 104, such as an NEC uPD 71071 Controller, on the buss handles memory to memory and serial I/O to memory transfers. An HDLC controller 106, such as an NEC uPD 72001, connected to the RAM 104, processes the data link layer (layer 2 of the OSI model). The DMA and HDLC controllers 104, 106 together operate at LAP-D protocol, carrying out conversion of data in parallel bit format to serial bit format for supply to ACFA 110

and IPAT 112. The ACFA 110 and IPAT 112, which are commercially available ISDN chips, such as ______ and ______, provide physical interfacing with the ISDN, performing framing, electrical signal conversion, and D channel collision detection functions. Using the RJ-11 plug shown in Figure 7 at the rear of the coprocessor board, the ACFA 110 is interfaced to an analog telephone 126 through a CODEC 128

The HDLC 106 controller has four channels, with one line incoming and another outgoing for each channel for full duplex operation. Logic implemented in the HDLC controller 106 identifies start and stop bits of each incoming block.

Upon receipt of a block as determined by the start and stop bit framing patterns, and following error detection using a checksum protocol indicating the block contains no errors, the HDLC controller 106 instructs the DMA controller 104 to pass the block to a selected location in a random access memory (RAM) 114 on the buss. If there is an error in the block, an error routine is executed.

Upon transmission of data, start and stop flags are applied to define a block, bit stuffing is carried out as necessary, and the block is discharged on the two B-channels comprising the BRI. Another HDLC controller 118, operating with a DMA controller 120, carries out similar functions for the D-channel of the ISDN line. Communication between the DMA controller 120 and the RAM 114 in the preferred embodiment is implemented using the SIP protocol of the commercially available Telios B101 ISDN communication coprocessor board.

Also residing on buss 100 is a UART 122 forming a COM port that interfaces data bidirectionally, one

block at a time, between buss 100 and a buss 124, that preferably is an ISA (AT compatible) buss, upon which resides the user's personal computer.

Another interface to user buss 124 is a memory 128 that is shared by both busses 100 and 124. The shared memory 128 is controlled by a shared memory controller 126 divided into two blocks, one for control information and the other for utilities such as debugging. The shared memory 128 receives data at separate read/write ports in a known manner to provide service to both of the busses 100 and 124.

Each program module, described below, will support message passing interfaces to and from other processes in the system and is an event driven, dispatchable thread being run under the real time kernel.

4. Platform Software

Figure 2, depicting symbolically the software architecture for implementing the computing platform, provided in the preferred embodiment by the Telios communication coprocessor system, provides the principal program modules and their interfaces resident in the system. An off-board PC Comm Package 210 together with UART 212 on the user buss 124 is coupled to the communication coprocessor circuit shown in the FIGURE at board resident UART 214. User side ISDN D channel signalling protocol is carried out at the data link layer (LAP-D as specified in Q.921) and at the network layer (Q.931) by software module 218. In the Telios B101 coprocesesor board, module 218 is implemented by a protocol termed "SIP", as mentioned earlier.

AT/PAD 216 multiplexes between the first and second B-channels forming the BRI by accessing

alternatively LAPD 1 and LAPD 2 modules 220 and 222 together with corresponding HDLC modules 224 and 226. These modules enable a separate data link connection over each of the B-channels in the system. The data link process implements full duplex operation using separate transmit and receive queues and interrupt driven thread dispatching based on the event of a message arriving to the system or a message being queued from transmission. These processes are implemented in the Telios coprocessor board as the V.120*/LAPD1 and V.120**/LAPD2 processes.

Shared memory manager module 230 directs flow of data between the shared memory 232 and the three BRI channels through interface 218 and HDLC 228 (D-channel) and LAPD, HDLC modules 220-226 (B1- and B2-channels).

The architecture further supports an analog telephone, managing conversion of analog to digital signals according to the ISDN standard including D channel signalling (SIP) protocol. This is termed the VFI processes in the Telios B101.

5. <u>Dynamic Channel Allocation</u>

Referring to FIGURE 3, the preferred embodiment of the invention is provided as two new program modules 302 and 304 to provide channel allocation management and virtual channels of greater bandwidth than a single B channel. For convenience, the description of these modules is provided with respect to the Teleos B101 ISDN Communication Coprocessor System, but the concept can be applied to any system supporting these new program modules. Program modules other than the new modules 302, 204 in Figure 3 correspond to like modules in FIGURE 2.

Module 302, the Channel management module (CMM),

monitors B channel usage and preempts the use of a B channel by a lower priority user when the channel is needed by a higher priority user. To accomplish this function, CMM 302 maintains a state table 502, shown symbolically in FIGURE 5(a), which indicates the state of each B channel including whether the channel is a "preemptable channel", that is, whether the channel is connected or disconnected and if connected, whether it is being used by a process that could release the channel in the event of preemption by a higher priority device. This process is carried out in the invention by Virtual Channel Process, described in more detail hereinafter, when more than one channel as part of a virtual channel is requested. The state table also records if a channel is in use as a result of preemption.

As shown in Figures 3 and 4, information necessary to maintain the table comes from messages sent by the signalling interface module 218 (SIP processes in the Telios communication coprocessor) to the CMM; messages are sent each time module 218 receives a request for channel connection or disconnection. Local requests originate from a user process running on the computer via shared memory interrupt manager 230, the Virtual Channel Process via internal message passing, AT/PAD processes via UART 214, or the Telios VFI process via Analog Phone Support 221. Remote requests arrive over the D channel. Upon receipt of such a message, the CMM 302 updates the state table 502 to record the new state of a B channel.

The CMM 502 also includes a table 504, shown symbolically in FIGURE 5(b), which maintains priority levels P(n) for all the end users n in the system. In the event that the CMM detects contention for a B

channel, i.e., there is a request to connect a new channel but all the channels are in use, the following additional action takes place.

Referring to the flowchart of FIGURE 6, and the message interface diagram of FIGURE 4, the CMM 302 first determines if a request for channel connection is received (step 600). When a request arrives, the module 302 determines whether the request is from an end user with greater priority than the users that are currently occupying the channels (step 602). If the request is from a user of equal or lower priority, the CMM takes no action and the signalling interface process 218 will deny the request (step 604). In the event that the request is from a higher priority user, the CMM 302 determines if a preemptable channel exists (step 606). If such a channel exists, the CMM will signal the process using the channel to logically free the channel from use (step 608), and when CMM 302 receives a signal indicating that the channel is no longer in use, step 610, the CMM will send a message to the signalling interface which will cause the channel to be disconnected (step 612). In this manner a channel will be made available to the higher priority end user. An example of this occurs when the Virtual Channel Process is using both B channels and the end user requesting a new channel is the analog telephone.

In the event that the CMM 302 receives a message indicating that a channel has been disconnected, in step 614, the CMM determines if the channel was a preempted channel (step 616). In this case the CMM will signal the process which had the channel preempted to reconnect the channel (step 618).

The virtual channel module (VCM) 304 provides the

capability to dynamically combine multiple, circuit-switched B channels into a single virtual channel which is capable of carrying a properly ordered sequence of data blocks. The VCM 394 comprises a block multiplexer/demultiplexer with buffering support to overcome the lack of synchronization across B channels inherent in the ISDN network. As shown in Figure 4, VCM 304 operates in conjunction with the CMM 302 to allocate and deallocate B channels based on events monitored by the CMM 302. The VCM 304 has a block oriented interface to the computer processes which use block send and block receive commands via the shared memory interface 126 of the communication coprocessor circuit shown in FIGURE 1. The computer process views the virtual channel as an ordered stream of blocks and has no knowledge of on which B channel the block is actually sent or received. The only indication to the computer process that the virtual channel is in use is the speed (bandwidth) available for transmission; as more channels are added or deleted, the bandwidth increases or decreases correspondingly.

The VCM 304 implements a protocol which operates as an upper sublayer of the data link layer, similar in purpose to but different in implementation from the ISO Multiple Link Procedure. This aspect of the invention is significant, as no processing at the physical layer, together with disadvantages associated therewith, takes place.

6. <u>Virtual Channel Protocol</u>

6.1 <u>B-Channel State Transitions</u>

Component channels allocated for a virtual B-channel have three defined states, viz., activated,

deactivated, and transient states. In the deactivated state, data transfer is forbidden although control protocol exchange is not. Component channels that have been allocated but not joined to the virtual channel are initially in the deactivated state. A channel is deactivated for a sender immediately after the deactive b command, described later, has been queued for transmission. For a receiver, a channel is deactivated immediately after the command is received. Only a bi-directionally deactivated channel is removed and physically disconnected from the virtual channel of which it was a member.

In the activated state, both data and control protocol transfer are permitted. A channel is activated to a sender only after acknowledgement of the active b command has been received. For a receiver, a channel is activated immediately following receipt of the active b command.

The transient state is defined by guarding periods before acknowledgements are received from the far-end. No user data can be sent over a channel in the transient state.

A virtual channel can be in either of two states, transient and steady. A virtual channel is transient when any component channels are transient. If no component channel is in the transient state, the Virtual Channel is in the steady state.

6.2 <u>Virtual Channel Multiplexing</u>

In a virtual channel, data received for transmission is sequentially distributed, one packet at a time, in a round- robin fashion over the channels in active state. The remote virtual channel process receives and recovers data in the same round-robin

sequence.

To add and delete channels, protocol messages are exchanged by the sender and receiver which cause the virtual channel to enter the transient state. An active b command is sent to add a channel; a deactive b command is sent to delete one. When a virtual channel is in the transient state, the round-robin send and receive mechanism continues but the additional features are added. On the receive side, a trace-back timer is used to ensure that error conditions on a transient line will not block the virtual channel for a long period of time. On the transmit side, if no user data is available, SYNC vB protocol messages optionally are sent on active lines to ensure that the receiver will initiate the trace back timer.

The traceback timer, which is implementation by programming within the program module VCM, is based upon an "optimistic algorithm" that assumes that a transmission sent ultimately will be received by the transmission partner. It is started at the probable logical end of transmission for recovery of time-inverted propogation of physical events. In the preferred embodiment, the traceback timer is iniated upon transmission of a block. If the communication partner does not respond within a predetermined amount of time, determined by the timer, the transmission is deemed to be invalid, and the channels are torn down and reinitiated. The operation of the traceback timer within the present system, depicted in the flow chart of Figure 10, is described in detail later.

Rather than global sequence numbers, a packet generation number preferably is used and has meaning only in terms of round-robin line scanning. Thus if two channels are in use, the first two packets will be

sent with a generation number of 1, the second two packets with a generation number of 2, etc. The sending and receiving processes both keep track of generation numbers and both use the same round-robin ordering so as to provide correct sequencing of packets.

6.3 Protocol Fields and Commands

The protocol commands, together with detailed format and bit definitions used for controlling and multiplexing over the virtual channel, are as follows.

* DATA

Ossnnnnn: one byte as header of client's data for transmit.

* RQST Bs

101nnnnn: one byte header of one or multiple control bytes. The first control byte can further specify explicitly whether the data is global. If the header does not specify, whether the data is global depends on the address bytes which follow. No address byte for in-band operation is required, and multiple address bytes imply that the in-band line is not explicitly included. The explicit global commands have the leftmost bit on, followed by explicitly specified address bytes. If the left- most two bits are both on, the data is a broadcast and no address bytes follow. Multiple bytes are useful for off-band control. Preferably, only one byte of in- band control is employed. The leftmost third bit is reserved, and the fourth bit is for system control point specification. These are external or internal indicated by E and I respectively as follows. The last

four bits define the commands of this category. At present, these are: DEACTIV Bs group: uwx1z, w bit is reserved and u defines two system points.

DEL Bs - (0X00 ¦ 0), an external delete command

DEX Bs - (DEL Bs ¦ 1), an DEL Bs, followed by external setup

DIL Bs - (DEL Bs ¦ 0X10), internally initiated delete command

DIX Bs - (DEX Bs ¦ 0X10), DIL Bs, followed by internal setup

ACTIV Bs group: uwx0z, w bit is reserved, u the same as defined above.

ADD Bs - (0X00 ¦ 2), an external add command.

AID Bs - (ADD Bs ¦ 0X10), an internally initiated ADD Bs

* SYNC vB

110nnnnn: one byte only, used for time-fill and synchronization.

* RA Bs

111nnnnn: one byte header, followed by no or multiple optional bytes. This is an acceptance acknowledge in reply to a RQST Bs, and in-band is implied if no other bytes follow. Similar to the case in RQST Bs, the first by which follows can further specify whether it is global. The left- most two bits have exactly the same meanings, which describe the scope of the possible additional address bytes. The leftmost third bit, if on, turns the whole meaning of the response into a negative acknowledgement. The last five bits are the same as

for RQST Bs.

* MODE vB
1000000m: for virtual channel operation and format specification. This has two members, MODE rst with m = 0 and MODE lbc with m = 1; MODE rst defines the nnnnn five bit generation number. The mode negotiation can proceed only when all component channels are in deactivated states, or UNA will be received. if the optional attributes are not accepted, UNA will also be received. A REJ will be received if the remote end does not support or will not accept this request. In addition to this role of mode negotiations, MODE vB resets the generation number or the global sequence number for transmitter and the corresponding variable back to 0 to restart and resets the transmitter and receiver (loop) pointers back to the first component position.

* RR
10000010: Informs the far-end receiver that this end is ready.

*

10000011: Informs the far-end receiver that his end is not ready.

* ATTRB vB
100000100: The parameters in APPENDIX I follow this for far-end negotiation.

* UA

10010000: This unnumbered positive response means acceptance in the mode negotiation. The same MODE vB packet will be sent back as bi-direction initialization, following the UA. The UA is also used for acknowledgement to flow control commands RR and RNR. It is used for acceptance acknowledgement to ATTRB vB as well as unnumbered commands.

* UNA

10010100: a negative response for refusing, used similarly to UA.

* REJ

10011100: as explained in the MODE vB description.

All those unnumbered are in-band, although MODE vB is for global. The "nnnnn" is a module 32 correlation generation number assigned to each round of transmissions. The "ss" in the DATA command, which has four values 11, 10, 01 nd 11, is for frame segmentation.

6.4 <u>Protocol Processing</u>

Programming described with reference to Figures 9 and 10 is provided for the two key routines in the Virtual Channel, Send_To_Remote and Receive_From_Remote. Both routines are initiated when a message is received from another process in the system, as shown in FIGURE 4.

Process: Send_To_Remote

Entry Condition:

This task executes when either a message arrives from another process requesting that a block of data be sent on the virtual channel (VC), or when the process itself must transmit messages due to virtual channel protocol processing, shown in FIGURE 4.

Data Structures Assumed:

1. Circular List of Transmission Lines maintained in the Round- Robin order of transmission.
2. Current Line Ptr.
3. VC Header block with current generation number.

Figure 9 is a flow chart describing the send_to remote routine in accordance with an aspect of this invention. Assume initially that a message is received on a channel from a remote source, represented by step 900. A decision is made on whether the virtual channel currently is in a steady state, i.e., not active (step 902). Assuming that the channel is in steady state, step 904 queries whether a data block is ready to send, i.e., a full block of data has been assembled. Only when a data block is ready, step 906 adds a virtual channel header to the data block, sends the block to a current line LAPD process, and then updates the current line to the next line in round-robin fashion. When a complete round of lines has been completed (step 908), the current generation number of the virtual channel header is updated in step 910.

If step 902 determines that the virtual channel is not in a steady state condition, step 912 of the program determines whether the current line is active. If the current line is active, step 914 creates an

active channel. If the current line is not active, step 914 determines whether a data block is ready for sending. When a data block is ready, steps 916, 918 and 920 carry out block assembly by adding a virtual channel header the block to be sent, then sending the block to the current line LAPD process, updating the line in round-robin fashion and finally updating the virtual channel header generation sequence number as in steps 906-910.

When step 914 determines that a data block is not ready to send, step 922 queries whether the transient line has any data to send. If there is not data to send, step 924 updates the current line in round-robin fashion. If data is available for sending, step 926 prepares data for transmission by carrying out a synchronization protocol, i.e., sending fill blocks with the virtual channel header, and then updates the header generation sequence number.

Pseudo code corresponding to the above routine is shown in Appendix II.

Process: Receive_From_Remote
Entry Condition:

This task executes when either a message arrives from the data link layer indicating that a new block has arrived from the Remote VC partner, or when the local Send_To_PC indicates that it is ready to receive another block.

Data Structures Assumed:

1. Circular List of Message Queues kept in the round-robin order of transmission.

2. Current Queue Ptr.

3. VC Header block with current generation number.

Protocol for receiving channel data from a remote source, in accordance with an aspect of this invention, can be better understood with reference to Figure 10. Assume initially that a message is received from a remote source (step 1000). Step 1002 determines whether the virtual channel is in steady state. Assuming that the virtual channel currently is in steady state, the message to be sent is placed in queue for the appropriate channel (step 1004). In step 1006, the queue is interrogated in a round-robin sequence to determine whether it is empty. When the queue is not empty, step 1008 processes the queue in round-robin order and sends the message to the appropriate party. This sequence of steps is repeated until a complete round of lines has been finished, determined by step 1010, and then the virtual channel header generation sequence number is updated (step 1012).

If the virtual channel is determined in step 1002 not to be in steady state, the traceback timer, noted previcously, and which preferably is software implemented, is queried to determine whether it is triggered (step 1014). Assuming that the timer has not yet been triggered, step 1016 detects whether the current line is empty. Only if the current line is empty, and assuming also that a non-empty line currently exists, and the generation number of messages on this line is current or future, determined by step 1018, the timing by the traceback timer is initiated. The program exits to step 1000 when either a message on the transient line arrives, or the traceback timer

expires (step 1020).

If step 1014 determines that the traceback timer is not triggered, step 1022 places a new message for the appropriate channel in the queue. If there is a message on the transient line, and the traceback timer is set, step 1024, the traceback timer is disabled (step 1026). If there is no message, the queue is queried on a round-robin basis to determine whether it is empty (step 1028). Assuming that the queue is not empty, the line is processed in round-robin order, step 1030, and the generation number of the message is read at step 1032 to determine whether it is current. If it is current, the message is delivered to the appropriate party (step 1034), if not, the message is discarded in step 1036.

On the other hand, if the queue on a round-robin basis is not empty, the virtual channel header generation sequence number is updated (step 1038). Pseudo code corresponding to the the above routine is given in Appendix III.

6.5 <u>Dynamic Channel Control</u>

The computer program which is using the Virtual Channel controls the number of B channels which can participate in the virtual channel by sending commands to the CMM module 302 to indicate that a channel should be added to or deleted from the virtual channel.

Channel allocation/deallocation can also occur based on actions by CMM 302 as described earlier. In the event that a channel is deallocated and then disconnected by the partner who was the calling party (i.e. the partner who did not originate the call), it is advantageous to provide a mechanism to indicate that the original party called should reestablish the

channel. The virtual channel protocol supports this by using control messages to indicate to the communication partner that the preempted channel is available again.

Using the routine shown in FIGURE 8, the VCM 304 deletes a B-channel if a high line error rate is detected. A predefined threshold rate of errors in a given period of time is provided to the VCM in step 802. If the data link layer for a given B-channel reports an error rate above this threshold, determined by step 804, the VCM 304 logically removes the B-channel from the virtual channel (step 806) and requests the CMM 302 to have the channel disconnected (step 808).

The invention described herein accordingly carries out dynamic channel allocation in an ISDN line by monitoring events supplied to the line which indicate requests to use or release a B channel, evaluating channel usage priorities associated with users generating events monitored, and, based upon channel usage priorities, automatically setting up or tearing down B channels associated with the line. Processing takes place above LAP-D protocol. Upon a communication coprocessor platform are provided first and second program modules, a channel management module (CMM) and a virtual channel module (VCM), to implement dynamic channel allocation. The program modules cooperate to control bandwidth between communication partners by selective allocation and deallocation of virtual B channels between them in response to preassigned priorities and real time events. The CMM monitors requests to connect and disconnect B channels by potential end users and the VCM. The CMM is capable of signalling the VCM to force deallocation of a B channel from a virtual channel based on a request for

channel connection by a higher priority end user. The CMM signals the VCM to indicate the existence of available B channels when end users disconnect the channel. The VCM provides a block oriented transport service to various computer processes which can operate over one or more B channels. This service operates above the data link layer and manages the multiplexing of messages across one or more data link connections which operate on a B channel so as to ensure properly ordered delivery of message blocks. Channels are added or deleted on a dynamic basis, without interrupting the flow of data. Channel deallocation also takes place in response to high error rates in message transmission.

In this disclosure, there is shown and described only the preferred embodiment of the invention, but, as aforementioned, it is to be understood that the invention is capable of use in various other combinations and environments and is capable of changes or modifications within the scope of the inventive concept as expressed herein.

APPENDIX I

Virtual B Channel Attributes (C language)

```
typedef struct {
  long   err_threshold;          /* one out of err_threshold frames *
  short  err_action_thres;       /* act on contig err_action_thres */
  short  err_action_permit;      /* allow deactivation of errored channel *
  short  max_band_width;         /* max allowed bandwidth */
  short  high_util_thres;        /* high effective % band width thresold *
  short  high_busy_util_thres;/* one high util out of high_busy_util*
  short  high_action_thres;      /* act on contig high_busy_util_thres *
  short  high_action_permit;     /* allow bandwidth increment actions *
  short  min_band_width;         /* minimum allowed bandwidth */
  short  low_util_thres;         /* low effective % bandwidth thresold *
  short  low_busy_util_thres;  /* one low util out of low_busy_util *
  short  low_action_thres;       /* act on contig low_busy_util_thres *
  short  low_action_permit;      /* allow bandwidth decrement actions *
  short  set_up_retry_thres;     /* max contig retry before giving up *
  short  no_action_thres;        /* no action threshold time */
} VB_THRESHOLD;
```

Table I - Virtual B Channel Attributes (in C language representation

APPENDIX II

SEND TO REMOTE ROUTINE

```
    If (VC is in Steady State)
     If (Datablock is ready to send)
       { add VC Header to Datablock;
         send block to Current_Line LAPD Process;
         update Current_Line to Next_Line in Round-robin
fashion;
         If (finished complete round of lines)
            update VC Header Generation Sequence Number;
       }
      Else
        exit; /* no data to send */
Else /* VC in Transient State */
   If (Current_Line is active)
      {
      If (Datablock is ready to send)
      { add VC Header to Datablock;
        send block to Current_Line LAPD Process;
        update Current_Line to Next_Line in Round-robin
fashion;
        If (finished complete round of lines)
           update VC Header Generation Sequence Number;
       }
      Else If (Transient Line has nothing to send)
        update Current_Line to Next_Line in Round-robin
fashion;
      Else If (Transient Line has something to send)
        {
        send Fill blocks with VC Header;
        update VC Header Generation Sequence Number;
         }
       }
   Else /* Current_Line is Transient Line */
      send VC protocol message to create active
channel;
```

APPENDIX III

RECEIVE FROM REMOVE ROUTINE

```
Receive_From_Remote  IF (VC is in Steady State)
   { place new message on queue for appropriate channel;
       While (Processing all queues in Round-Robin order
            and queue is not empty)
           Deliver message to appropriate party;

       If (finished complete round of lines)
            update VC Header Generation Sequence Number;
   }
  Else (VC is in Transient State)
    If (traceback timer has not triggered)
     If (Current_Line is empty)
      {If (non-empty line exists) and
         (generation number of message on this line is current or future)
            { Start Traceback timer;
              Wait for message on the transient line or timer to expire;
            }
      }
     Else
      { place new message on queue for appropriate channel;
         If (message is on transient line and Traceback timer is set)
            disable Traceback timer;
    Else /* traceback timer has triggered)
      { While (Processing all lines in Round-Robin order)
        { if (generation number of message in queue is current)
              Deliver message to appropriate party;

            else if (generation number is past)
              discard message;
        }
        Update VC Header Generation Sequence Number
      }
```

CLAIMS:

1. In a computer interface to ISDN, a method of automatically allocating/deallocating B-channels, comprising the steps of above the LAPD layer:

(a) monitoring events supplied to the ISDN which indicated requests to use or release a B-channel;

(b) evaluating channel usage priorities associated with users generating events monitored in step (a); and

(c) based upon said channel usage priorities, automatically setting up or tearing down B-channels associated with said ISDN.

2. The method of claim 1, wherein said events include setup and teardown messages arriving to the ISDN over the D-channel of said ISDN.

3. The method of claim 1, wherein said events further include generation of local messages by said computer or other devices locally attached to said interface.

4. The method of claim 1, wherein said ISDN is a Basic Rate Interface (BRI).

5. The method of claim 1, including the steps of activating a virtual B-channel between two compatible computer processes using a B-channel set up in step (c), and wherein step (c) includes the steps of logically adding or deleting channels to or from said virtual B-channel, and tearing down said virtual B-channel based upon a condition predetermined by said computer processes.

6. The method of claim 1, wherein said events

include local events generated by an analog telephone interfaced by said computer interfaced to the ISDN.

7. The method of claim 1, wherein said computer interface includes at least one of a shared memory component or UART component, and wherein step (a) includes monitoring events generated by a computer process communicating with said shared memory or UART component of said interface.

8. The method of claim 1, wherein step (b) includes assigning properties to said events, and step (c) includes allocating or deallocating B-channels based upon bandwidth requirements of events having particular properties.

9. The method of claim 8, including activating a virtual B-channel between two compatible computer processes using a B-channel set up in step (c), and wherein said step of assigning is carried out during said step of activating.

10. The method of claim 1, wherein step (c) includes further allocating or deallocating B-channels complementarily.

11. In a computer interface to ISDN, a method of forming a virtual B-channel, for communication between two compatible computer processes, out of a pool of B-channels from said ISDN, comprising the steps of:

(a) activating a virtual B-channel with a B-channel from said ISDN;

(b) logically adding or deleting a B-channel into or from said ISDN into said virtual B-channel activated

in step (a);

(c) dynamically allocating or deallocating B-channels to or from said virtual B-channel based upon usage by said computer processes; and

(d) tearing down said virtual B-channel based upon a condition predetermined by said computer processes.

12. The method of claim 11, wherein step (c) includes a step of implementing a protocol between said computer processes for using all B-channels of said virtual B-channel while dynamically allocating or deallocating B-channels to or from said virtual B-channel based upon channel usage by said computer processes.

13. The method of claim 11, wherein said communication is in a full duplex mode.

14. The method of claim 11, including the additional steps of:

(a) monitoring remote events on the D-channel of said ISDN and local events to the ISDN; and

(b) in response to step (a), evaluating channel requirements of said remote or local events;

wherein step (b) includes the step of automatically setting up or tearing down a B-channel based upon said channel requirements.

15. The method of claim 12, wherein the step of implementing includes exchanging between said processes information sufficient to indicated that an incoming call set up request is directed to said virtual B-channel, and allocating a B-channel thereto.

16. The method of claim 11, wherein step (b) is initiated by either one of said processes.

17. The method of claim 11, wherein step (b) is initiated by the ISDN.

18. Apparatus for interfacing a personal computer to ISDN, comprising:

a processor; and

program means executable by said processor and operating above the LAPD Layer of the Open Systems Integration (OSI) model for monitoring events supplied to the ISDN which indicate requests to use or release a B-channel; evaluating channel usage priorities associated with users generating events; and based upon said channel usage priorities, automatically setting up or tearing down B-channels associated with said ISDN.

19. The apparatus of claim 18, wherein said program means includes means for activating a virtual B-channel between two compatible computer processes using a B-channel that has been set by said program means, and logically adding or deleting channels to or from said virtual B-channel, and tearing down said virtual B-channel based upon a condition predetermined by said computer processes.

20. The apparatus of claim 18, wherein said program means includes means for assigning properties to said events and allocating or deallocating B-channels based upon bandwidth requirements of events having particular properties.

21. The apparatus of claim 18, including at least

one of a shared memory component and UART component, and wherein said program means includes means for monitoring events generated by a computer process communicating with said shared memory or UART component.

22. Apparatus for interfacing a personal computer to ISDN, comprising:

a processor; and

program means executable by said processor and operating above the LAPD Layer of the Open Systems Integration (OSI) model for activating a virtual B-channel with a B-channel from said ISDN; logically adding or deleting a B-channel into or from said ISDN into an activated virtual B-channel; dynamically allocating or deallocating B-channels to or from said virtual B-channel based upon usage by said computer processes; and tearing down said virtual B-channel based upon a condition predetermined by said computer processes.

23. The apparatus of claim 22, wherein said program means includes means for implementing a protocol between said computer processes for using all B-channels of said virtual B-channel while dynamically allocating or deallocating B-channels to or from said virtual B-channel based upon channel usage by said computer processes.

24. The apparatus of claim 22, wherein said program means includes monitoring remote events on the D-channel of said ISDN and local events to the ISDN, and, in response, evaluating channel requirements of said remote or local events including automatically

setting up or tearing down a B-channel based upon said channel requirements.

25. The apparatus of claim 22, including at least one of a shared memory component and UART component, and wherein said program means includes means for monitoring events generated by a computer process communicating with said shared memory or UART component.

26. Apparatus for interfacing a personal computer to ISDN, comprising:

a processor; and

program means executable by said processor and operating above the LAPD Layer of the Open Systems Integration (OSI) model for activating a virtual B-channel with a B-channel from said ISDN, logically adding or deleting a B-channel into or from said ISDN into an activated virtual B-channel, dynamically allocating or deallocating B-channels to or from said virtual B-channel based upon usage by said computer processes, and tearing down said virtual B-channel based upon a condition predetermined by said computer processes;

said program means comprising

(a) a channel management module (CMM) for monitoring requests to connect or disconnect B-channels by users; and

(b) a virtual channel module (VCM) responsive to information from said CMM and providing a block oriented transport service to computer processes capable of operating over one or more B-channels for allocating or deallocating B-channels to or from a virtual B-channel without interrupting flow of data in

said virtual B-channel.

27. The apparatus of claim 26, wherein said CMM further supplies to said VCM information on availability of a B-channel when a user disconnects it.

28. The apparatus of claim 26, wherein said CMM further supplies to said VCM information instructing deallocation of a B-channel from said virtual B-channel upon a request for channel connection by a user having a priority greater than the priority of a current user.

29. The apparatus of claim 26, wherein said VCM includes means for detecting in a line a message transmission containing an error rate greater than a predetermined acceptable error rate, and in response, requesting deallocation of said line.

30. The apparatus of claim 26, wherein said program means further includes traceback timer means initiated at the probable logical end of a transmission for recovery of time inverted propagation of events on the ISDN.

31. The apparatus of claim 26, wherein said program means further includes means for operating a channel selectively in deactivated, activated and transient states.

32. The apparatus of claim 26, wherein said program means further includes means for synchronization channels.

33. The apparatus of claim 32, wherein said

synchronization means includes means for performing simultaneous longitudinal and transverse data flow regulation in said channels.

34. The apparatus of claim 33, wherein said longitudinal data regulation comprises regulation by generation number.

35. The apparatus of claim 33, wherein said transverse data regulation comprises regulation by a round-robin protocol.

1/10

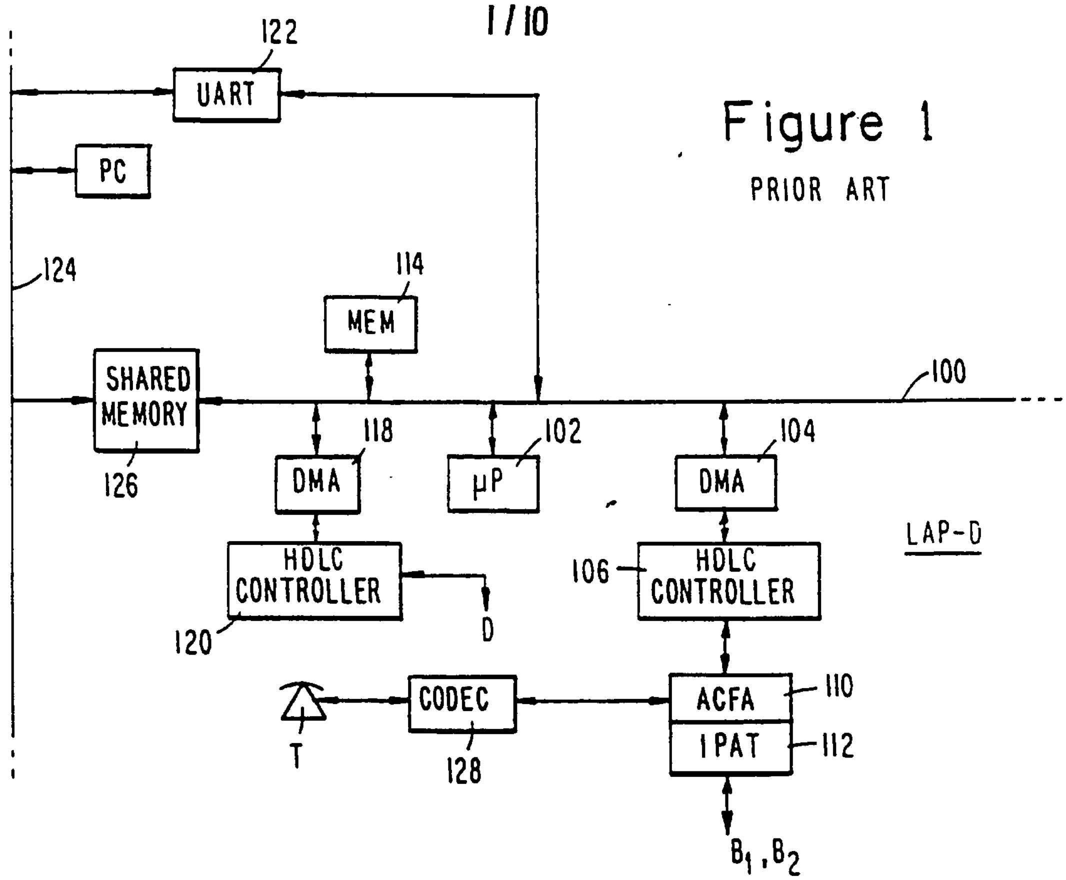


Figure 1

PRIOR ART

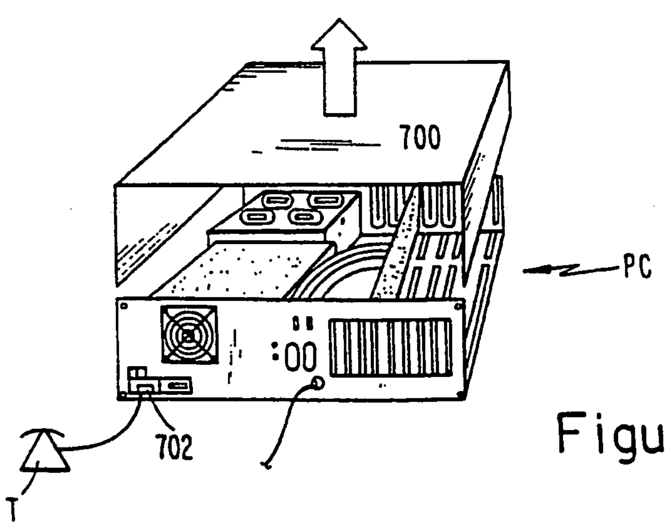


Figure 7

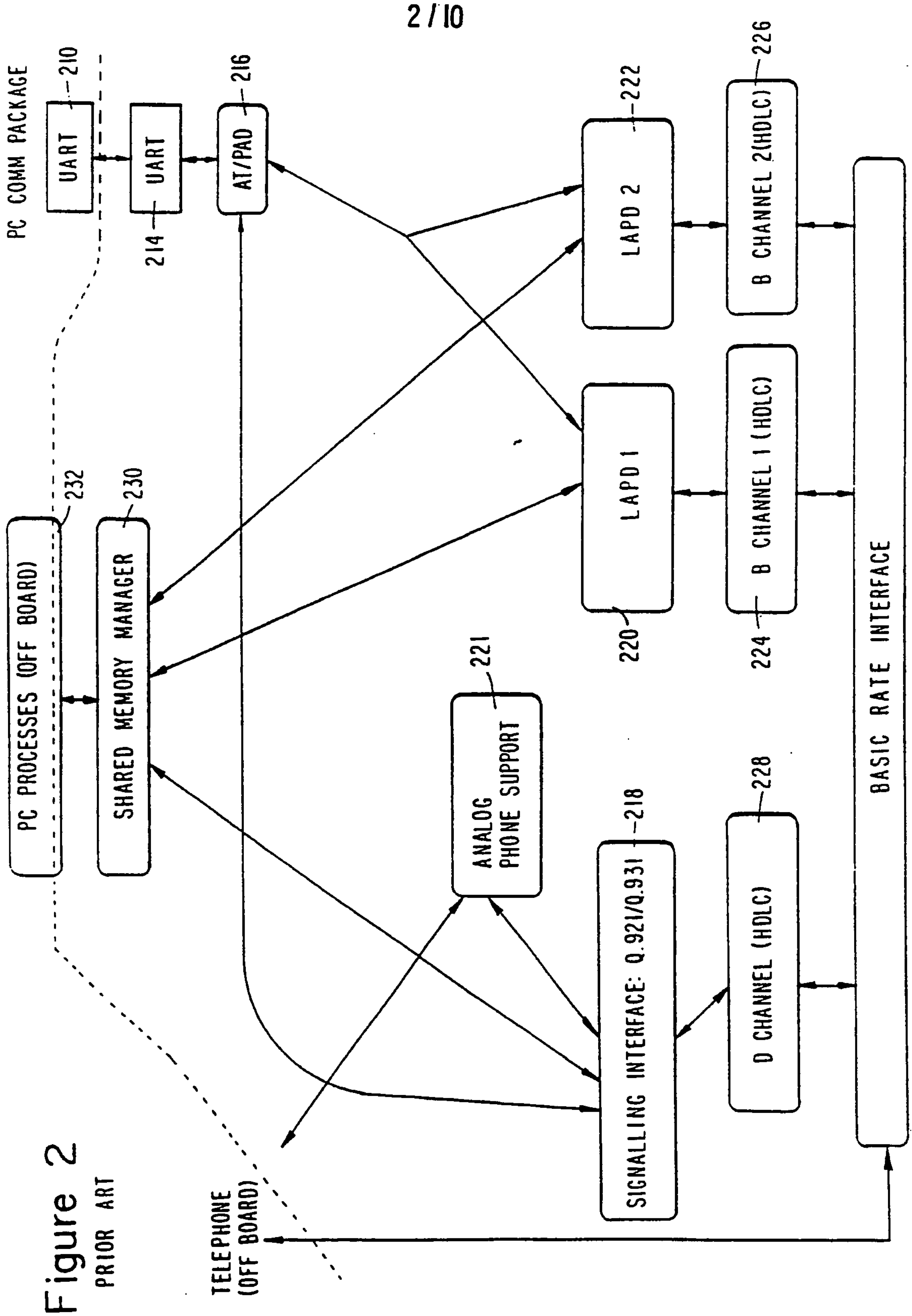


Figure 2
PRIOR ART

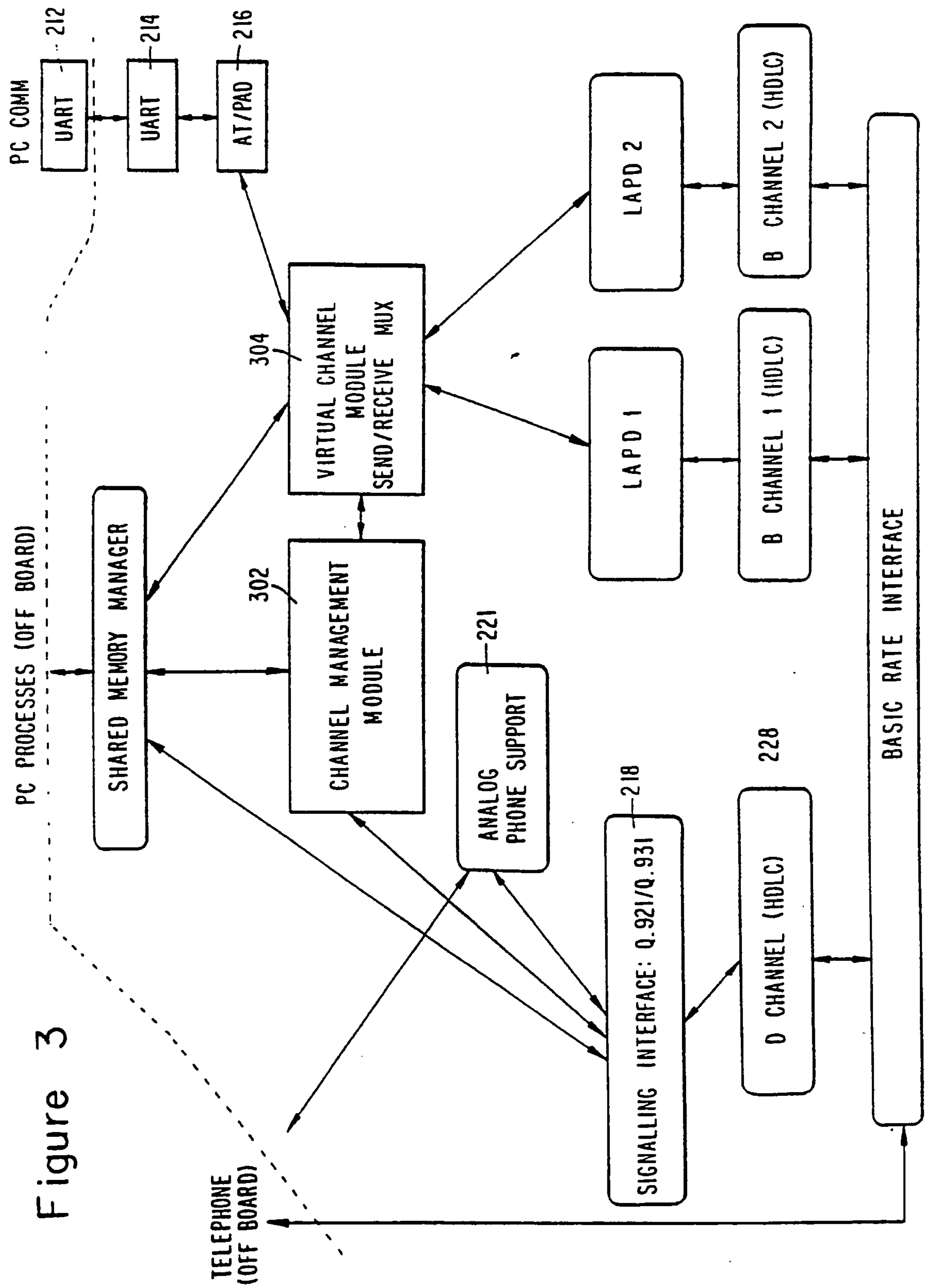

PC COMM
UART
212
UART
214
AT/PAD
216
PC PROCESSES (OFF BOARD)
SHARED MEMORY MANAGER
304
VIRTUAL CHANNEL MODULE SEND/RECEIVE MUX
302
CHANNEL MANAGEMENT MODULE
LAPD 2
LAPD 1
B CHANNEL 2 (HDLC)
B CHANNEL 1 (HDLC)
221
ANALOG PHONE SUPPORT
218
SIGNALLING INTERFACE: Q.921/Q.931
228
D CHANNEL (HDLC)
BASIC RATE INTERFACE
TELEPHONE (OFF BOARD)

Figure 3

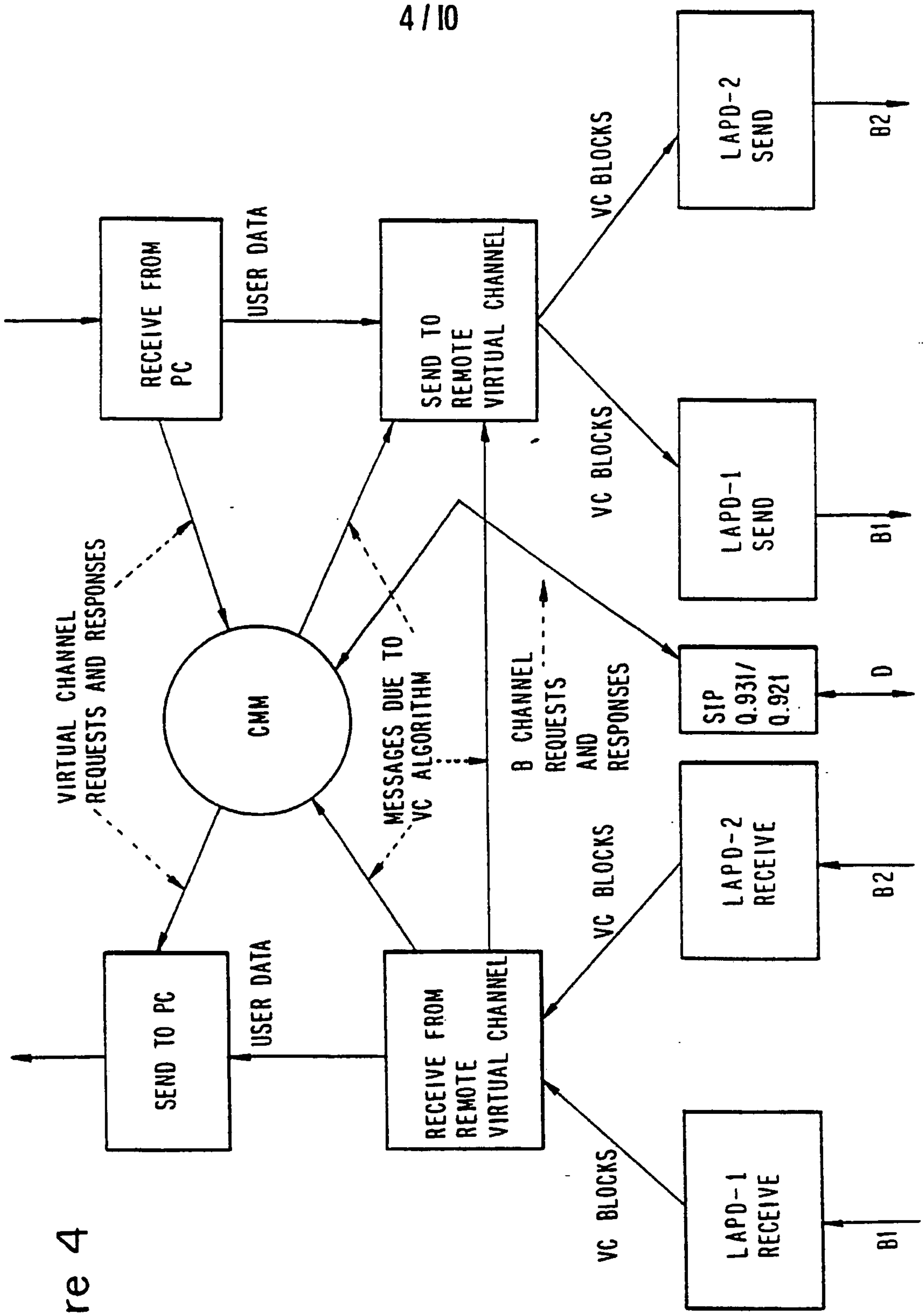

RECEIVE FROM PC
USER DATA
SEND TO REMOTE VIRTUAL CHANNEL
VC BLOCKS
LAPD-2 SEND
B2
VC BLOCKS
LAPD-1 SEND
B1
VIRTUAL CHANNEL REQUESTS AND RESPONSES
CMM
MESSAGES DUE TO VC ALGORITHM
B CHANNEL REQUESTS AND RESPONSES
SIP Q.931/ Q.921
D
VC BLOCKS
LAPD-2 RECEIVE
B2
SEND TO PC
USER DATA
RECEIVE FROM REMOTE VIRTUAL CHANNEL
VC BLOCKS
LAPD-1 RECEIVE
B1

Figure 4

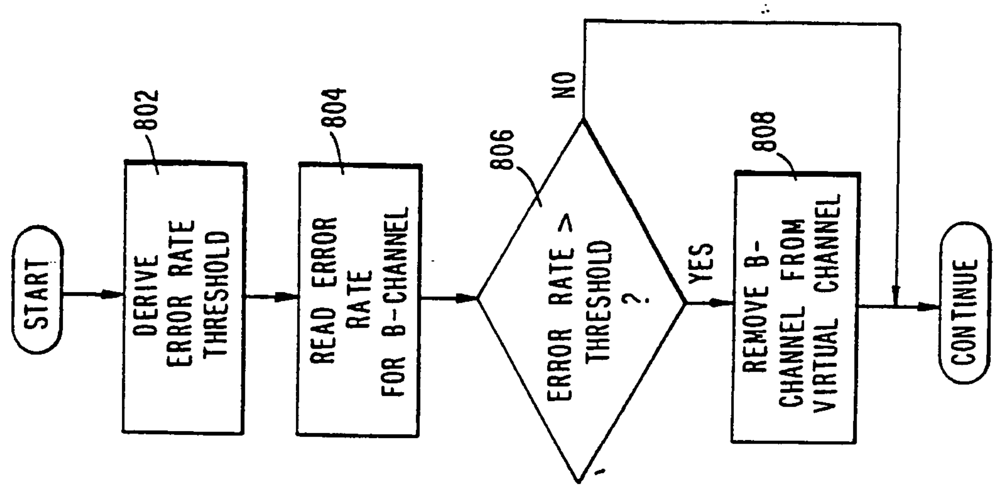


Figure 8

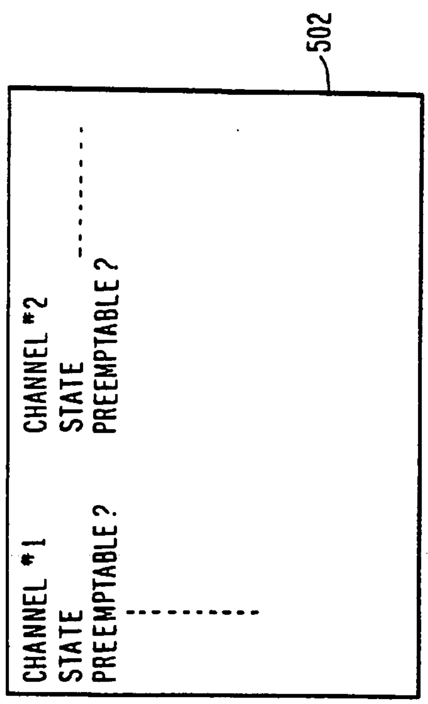


Figure 5(a)

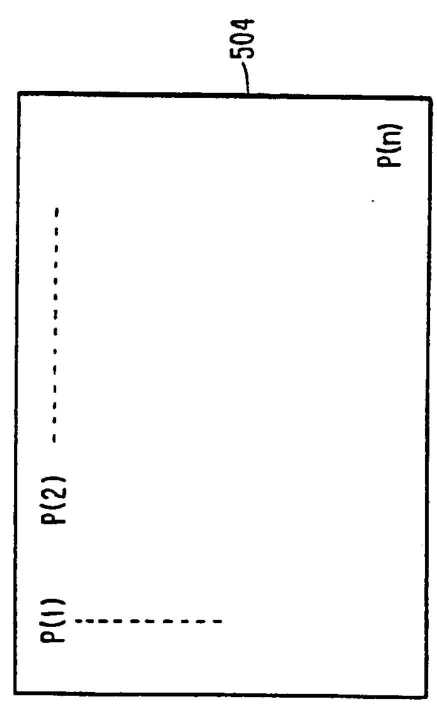
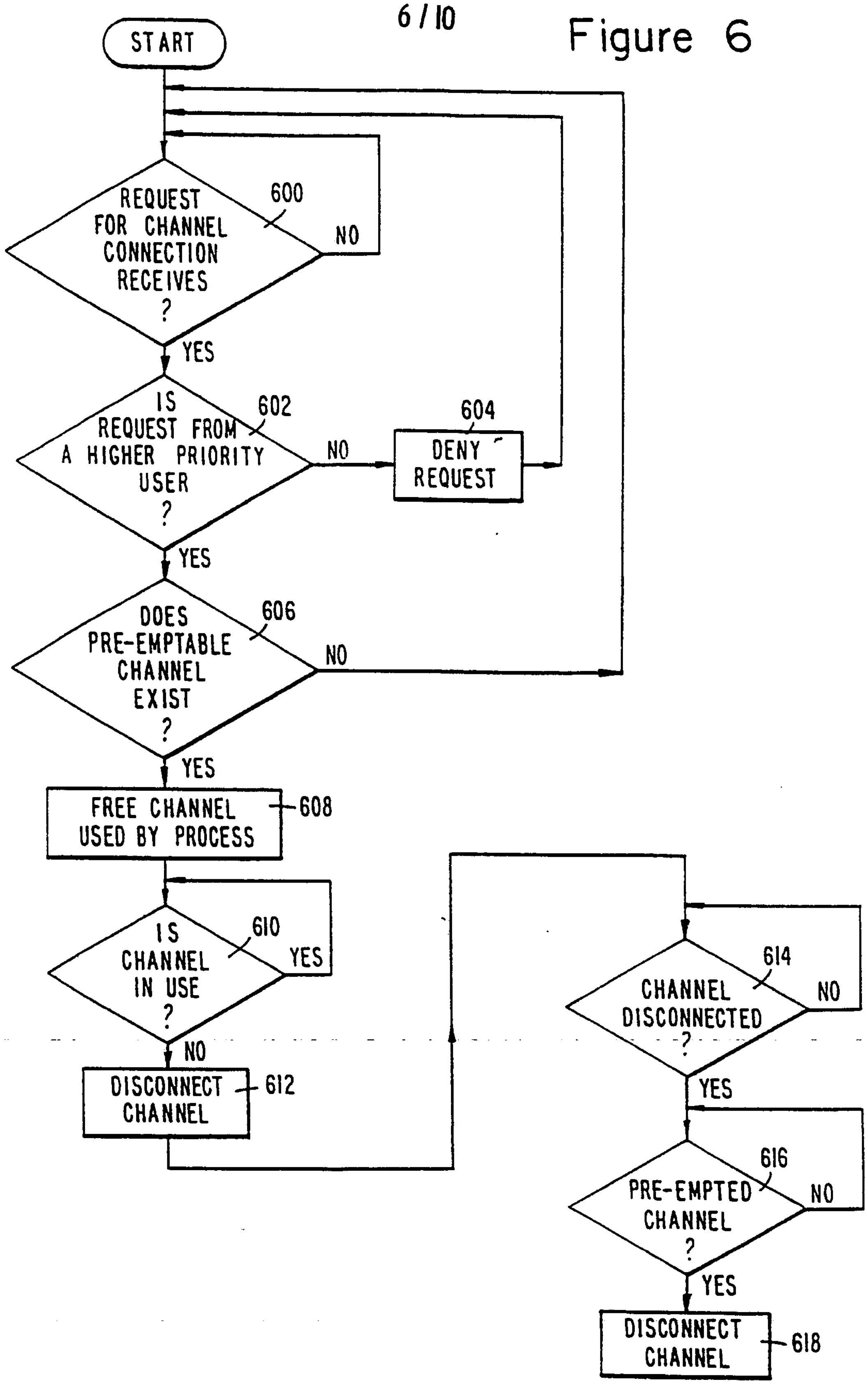


Figure 5(b)

Figure 6


START
REQUEST FOR CHANNEL CONNECTION RECEIVES ?
600
NO
YES
IS REQUEST FROM A HIGHER PRIORITY USER ?
602
NO
DENY REQUEST
604
YES
DOES PRE-EMPTABLE CHANNEL EXIST ?
606
NO
YES
FREE CHANNEL USED BY PROCESS
608
IS CHANNEL IN USE ?
610
YES
NO
DISCONNECT CHANNEL
612
CHANNEL DISCONNECTED ?
614
NO
YES
PRE-EMPTED CHANNEL ?
616
NO
YES
DISCONNECT CHANNEL
618

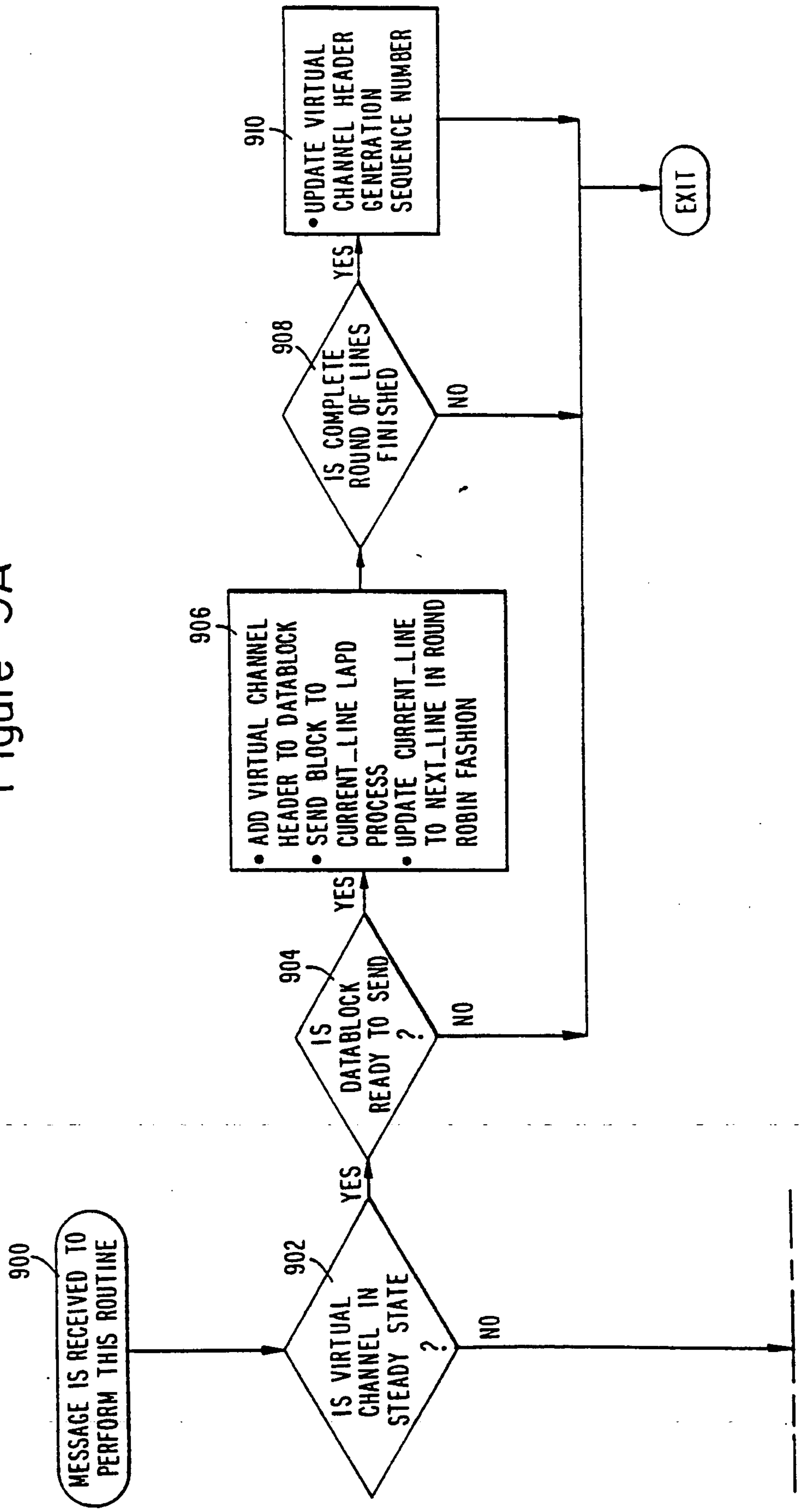

900
MESSAGE IS RECEIVED TO PERFORM THIS ROUTINE
902
IS VIRTUAL CHANNEL IN STEADY STATE ?
YES
NO
TO FIG. 9B
904
IS DATABLOCK READY TO SEND ?
YES
NO
906
• ADD VIRTUAL CHANNEL HEADER TO DATABLOCK
• SEND BLOCK TO CURRENT_LINE LAPD PROCESS
• UPDATE CURRENT_LINE TO NEXT_LINE IN ROUND ROBIN FASHION
908
IS COMPLETE ROUND OF LINES FINISHED
YES
NO
910
• UPDATE VIRTUAL CHANNEL HEADER GENERATION SEQUENCE NUMBER
EXIT

Figure 9A

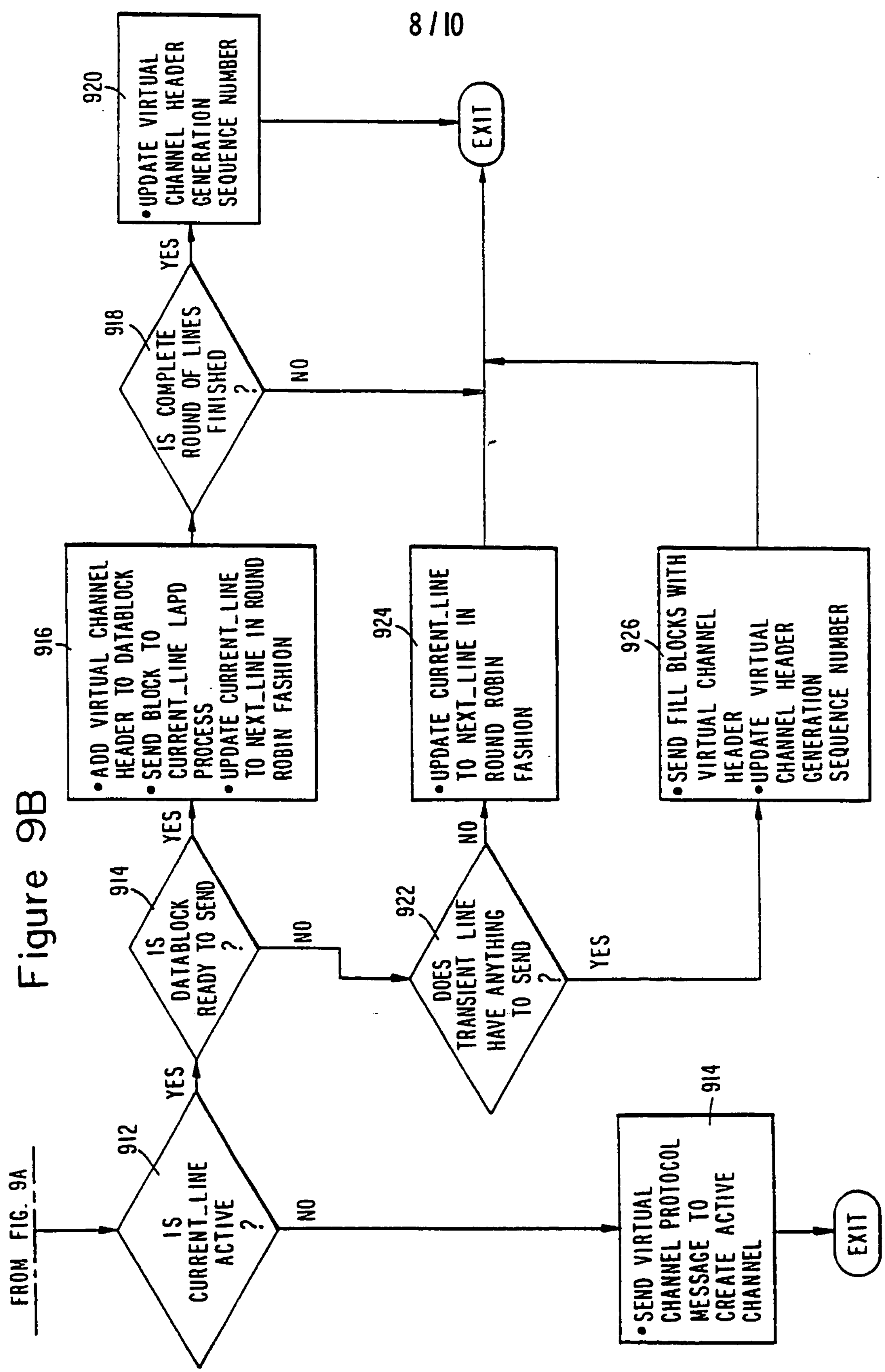
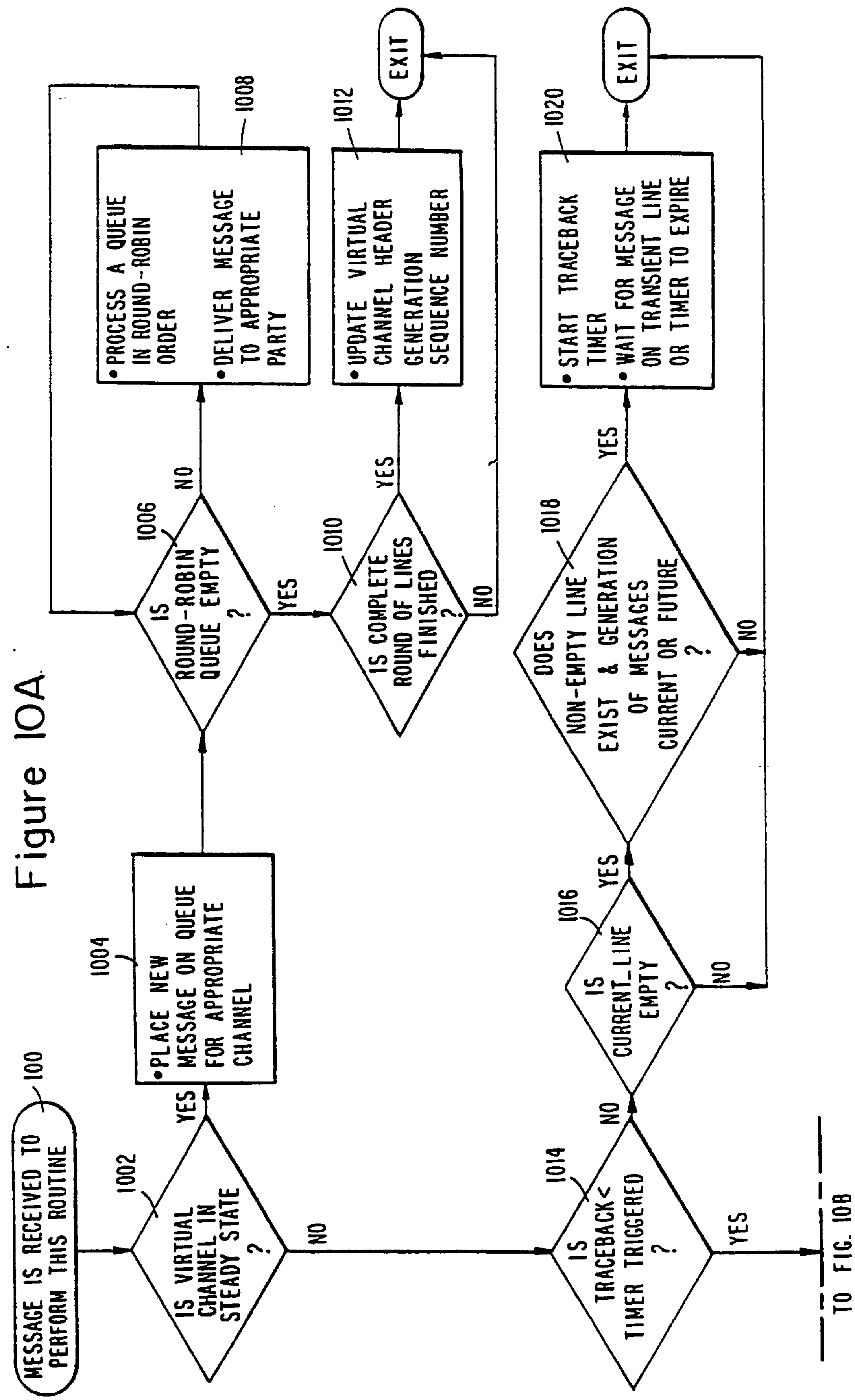

FROM FIG. 9A
912
IS CURRENT_LINE ACTIVE ?
YES
NO
914
IS DATABLOCK READY TO SEND ?
YES
NO
916
• ADD VIRTUAL CHANNEL HEADER TO DATABLOCK
• SEND BLOCK TO CURRENT_LINE LAPD PROCESS
• UPDATE CURRENT_LINE TO NEXT_LINE IN ROUND ROBIN FASHION
918
IS COMPLETE ROUND OF LINES FINISHED ?
YES
NO
920
• UPDATE VIRTUAL CHANNEL HEADER GENERATION SEQUENCE NUMBER
EXIT
922
DOES TRANSIENT LINE HAVE ANYTHING TO SEND ?
NO
YES
924
• UPDATE CURRENT_LINE TO NEXT_LINE IN ROUND ROBIN FASHION
926
• SEND FILL BLOCKS WITH VIRTUAL CHANNEL HEADER
• UPDATE VIRTUAL CHANNEL HEADER GENERATION SEQUENCE NUMBER
914
• SEND VIRTUAL CHANNEL PROTOCOL MESSAGE TO CREATE ACTIVE CHANNEL
EXIT

Figure 9B


MESSAGE IS RECEIVED TO PERFORM THIS ROUTINE
100
1002
IS VIRTUAL CHANNEL IN STEADY STATE ?
YES
NO
1004
• PLACE NEW MESSAGE ON QUEUE FOR APPROPRIATE CHANNEL
1006
IS ROUND-ROBIN QUEUE EMPTY ?
NO
YES
1008
• PROCESS A QUEUE IN ROUND-ROBIN ORDER
• DELIVER MESSAGE TO APPROPRIATE PARTY
1010
IS COMPLETE ROUND OF LINES FINISHED ?
YES
NO
1012
• UPDATE VIRTUAL CHANNEL HEADER GENERATION SEQUENCE NUMBER
EXIT
1014
IS TRACEBACK TIMER TRIGGERED ?
NO
YES
TO FIG. 10B
1016
IS CURRENT_LINE EMPTY ?
YES
NO
1018
DOES NON-EMPTY LINE EXIST & GENERATION OF MESSAGES CURRENT OR FUTURE ?
YES
NO
1020
• START TRACEBACK TIMER
• WAIT FOR MESSAGE ON TRANSIENT LINE OR TIMER TO EXPIRE
EXIT
Figure 10A

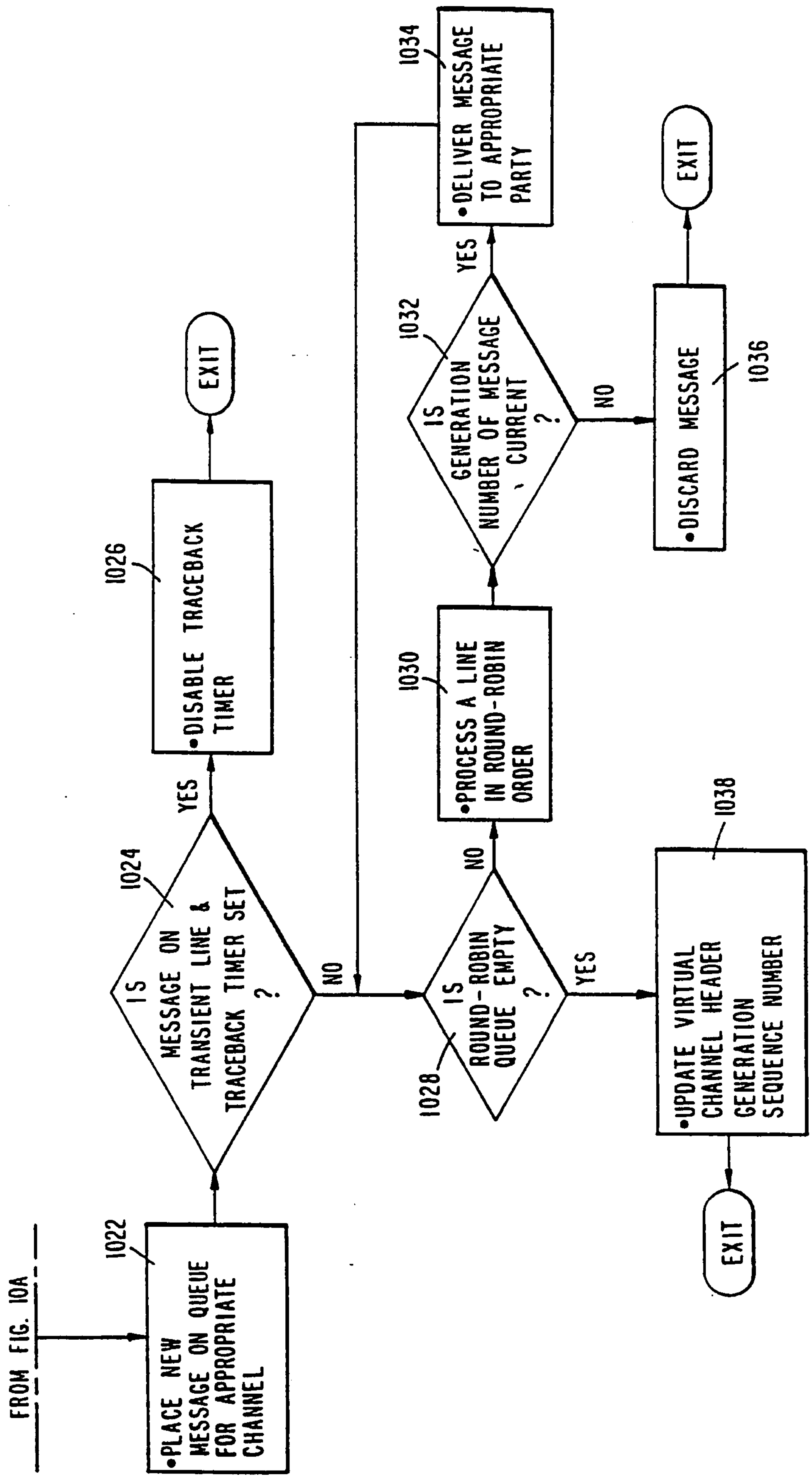

FROM FIG. 10A
1022
•PLACE NEW MESSAGE ON QUEUE FOR APPROPRIATE CHANNEL
1024
IS MESSAGE ON TRANSIENT LINE & TRACEBACK TIMER SET ?
YES
NO
1026
•DISABLE TRACEBACK TIMER
EXIT
1028
IS ROUND-ROBIN QUEUE EMPTY ?
NO
YES
1030
•PROCESS A LINE IN ROUND-ROBIN ORDER
1032
IS GENERATION NUMBER OF MESSAGE CURRENT ?
YES
NO
1034
•DELIVER MESSAGE TO APPROPRIATE PARTY
1036
•DISCARD MESSAGE
EXIT
1038
•UPDATE VIRTUAL CHANNEL HEADER GENERATION SEQUENCE NUMBER
EXIT

Figure 10B

INTERNATIONAL SEARCH REPORT

International Application No. PCT/US91/03448

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all) [6]

According to International Patent Classification (IPC) or to both National Classification and IPC

INT CL(5): H04J 3/12, 3/24
U.S. Cl.: 370/110.1, 94.1

II. FIELDS SEARCHED

Minimum Documentation Searched [7]

Classification System	Classification Symbols
US	370/110.1, 94.1, 94.2, 94.3, 60, 60.1

Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched [8]

III. DOCUMENTS CONSIDERED TO BE RELEVANT [9]

Category *	Citation of Document, [11] with indication, where appropriate, of the relevant passages [12]	Relevant to Claim No. [13]
A	US, A, 4,592,045 Fundneider, 27 May 1986 See entire document	1-35
A	US, A, 4,878,216 Yunoki, 31 October 1989 See entire document	1-35
A	US, A, 4,896,350 Bicknell et al, 23 January 1990 See entire document	1-35
A	US, A, 4,958,153 Murata et al, 18 September 1990 See entire document	1-35
A	US, A, 4,958,343 Abramovici et al, 18 September 1990 See entire document	1-35

* Special categories of cited documents: [10]

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

IV. CERTIFICATION

Date of the Actual Completion of the International Search	Date of Mailing of this International Search Report
10 July 1991	05 AUG 1991
International Searching Authority	**Signature of Authorized Officer**
ISA/US	Wellington Chin